EVALUATION OF ROS2 COMMUNICATION LAYER

Adam Dąbrowski, Rafał Kozik, Mateusz Maciaś
Industrial Research Institute for Automation and Measurements PIAP
Agenda

• Why we need robustness and resilience in ROS
• Introduction to ROS2 and DDS
• Evaluation methodology
• Results
• Conclusions
Why do we need robust and resilient ROS?
Data Distribution Service (DDS)
Quality of Service (QoS)

- Reliability
- History
- Durability
Supported DDS implementations

- RTI Connext DDS Wrapper
- PrismTech OpenSplice Wrapper
- eProsima Fast RTPS Wrapper
Evaluation environment

Docker

Container
ROS2 Node

VETH

3 topics with different QoS

Container
ROS2 Node

VETH

Virtual interface modified with tc-netem
Scenario 1

- Random packets are being dropped
- Presented results are for RTI Connext DDS
- Graphs show mean values from 28 runs
ROS reception rate in lossy networks

![Graph showing ROS reception rate (%) vs. Packet loss rate (%)](image-url)
ROS2: reception rate for critical messages

![Graph showing the relationship between packet loss rate and reception rate. The reception rate remains high until the packet loss rate reaches around 50%, after which it drops drastically.](image-url)
Participant liveliness

Participant 1

LIVELINESS MESSAGE

LIVELINESS MESSAGE

LIVELINESS MESSAGE

LIVELINESS MESSAGE

Participant 2

ASSERT PERIOD
ROS2: reception rate for control messages
ROS2: reception rate for sensor data

![Graph showing the relation between reception rate and packet loss rate. The graph indicates a linear decrease in reception rate as the packet loss rate increases.](image)
ROS2: latency for critical messages

RobotAlarm message: latency
ROS2: latency for control messages

![Graph showing latency in ms vs packet loss rate in %]
Scenario 2

- High volume of sensor data
- There is a limit on throughput
- Presented results are for eProsima Fast RTPS
- Graphs show mean values from 28 runs
ROS2: reception rate for critical messages
ROS2: reception rate for sensor data
ROS2: latency for critical messages
Differences in implementations
Limitations of simplified DDS API

• Missing Quality of Service settings:
  • Deadline
  • LatencyBudget

• No access to useful events:
  • Liveliness lost
  • Offered deadline missed

• Limited configuration
  • Participant liveliness
  • Heartbeat period
<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ROS2 makes use of the mature and powerful DDS standard</td>
<td>• DDS is a complex standard with ample configuration options</td>
</tr>
<tr>
<td>• Flexibility to choose a suitable implementation</td>
<td>• Differences between the implementations</td>
</tr>
<tr>
<td></td>
<td>• Limited access to the DDS features</td>
</tr>
</tbody>
</table>
Thank you!

Adam Dąbrowski  
linkedin.com/in/dabrowskiadam

Rafał Kozik  
linkedin.com/in/rafalkozik

https://github.com/piapplt/ros2_benchmarking